

Tehnike programiranja

PREDAVANJE 11

Predavanje 11

idos.fe.uni-lj.si

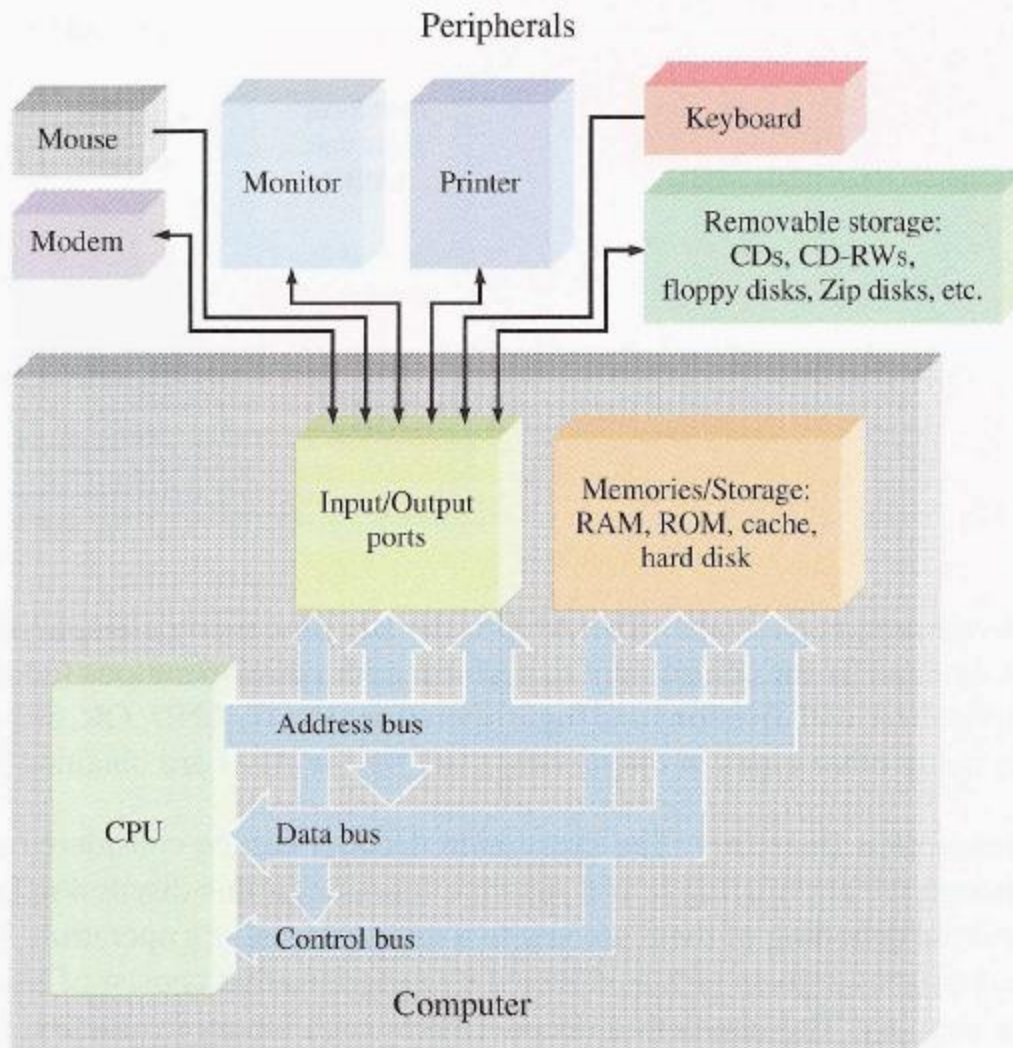
>študij >Tehnike programiranja

>Gradiva

- Načrtovanje programov
- Programski jeziki

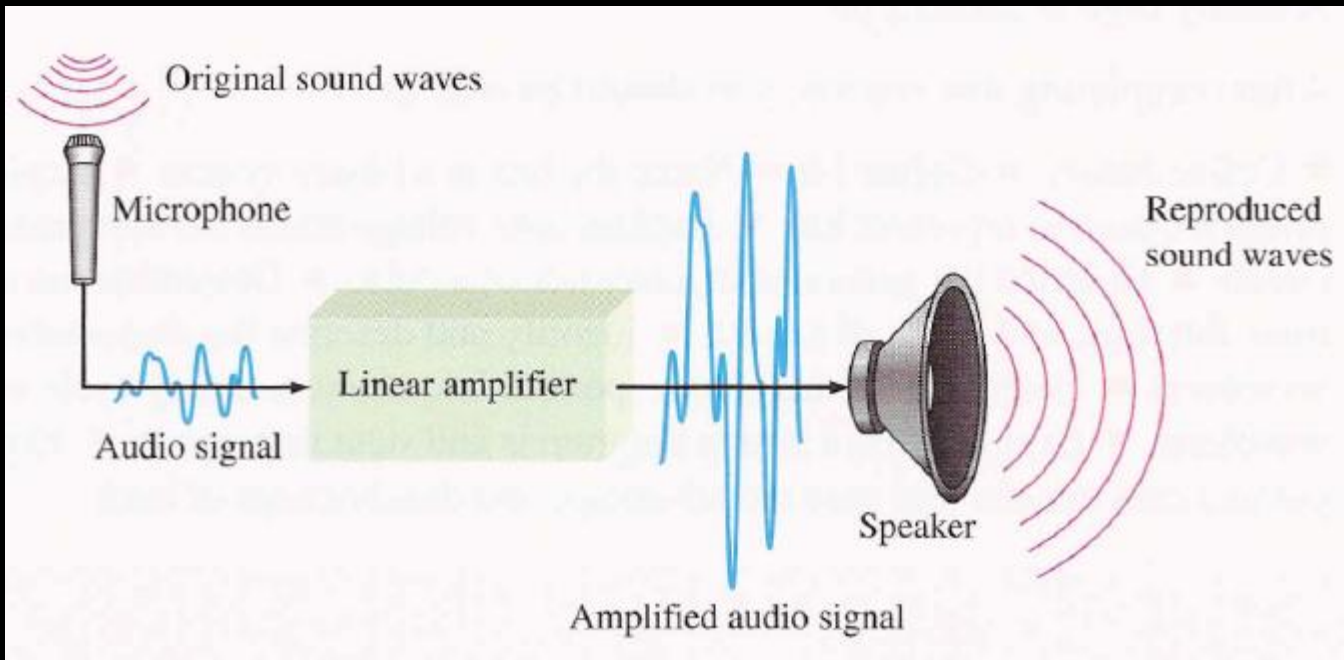
Kolokvij:

28.5. ob 9:30

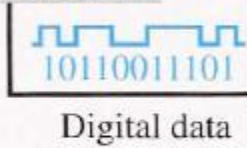


◀ **FIGURE 12-2**

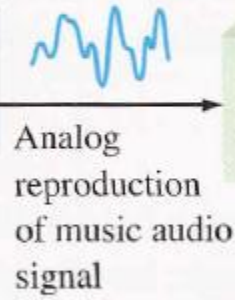
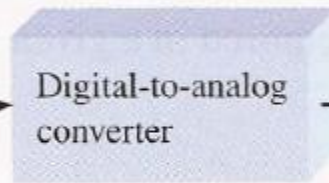
Basic block diagram of a typical computer system including common peripherals. The computer itself is shown within the gray block.



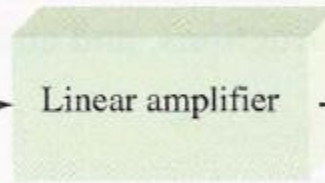
CD drive



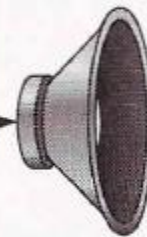
Digital data



Analog reproduction of music audio signal



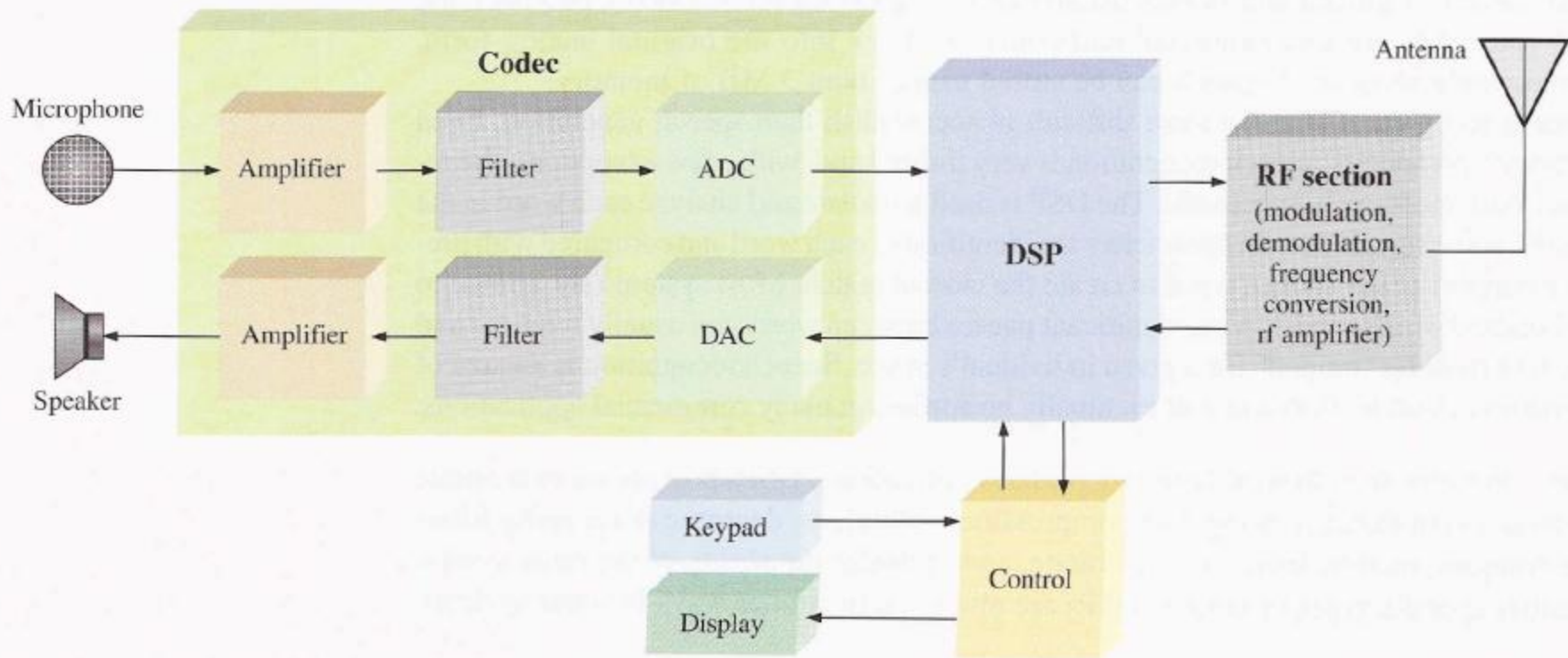
Linear amplifier



Speaker



Sound waves



▲ **FIGURE 13-29**

Simplified block diagram of a digital cellular phone.

Program

- algoritem – določeno zaporedje ali niz navodil, ki opisujejo rešitev problema
 - diagram poteka
 - sestavine programa

 - Primer: podaj algoritem za naslednji problem...
-

Programiranje

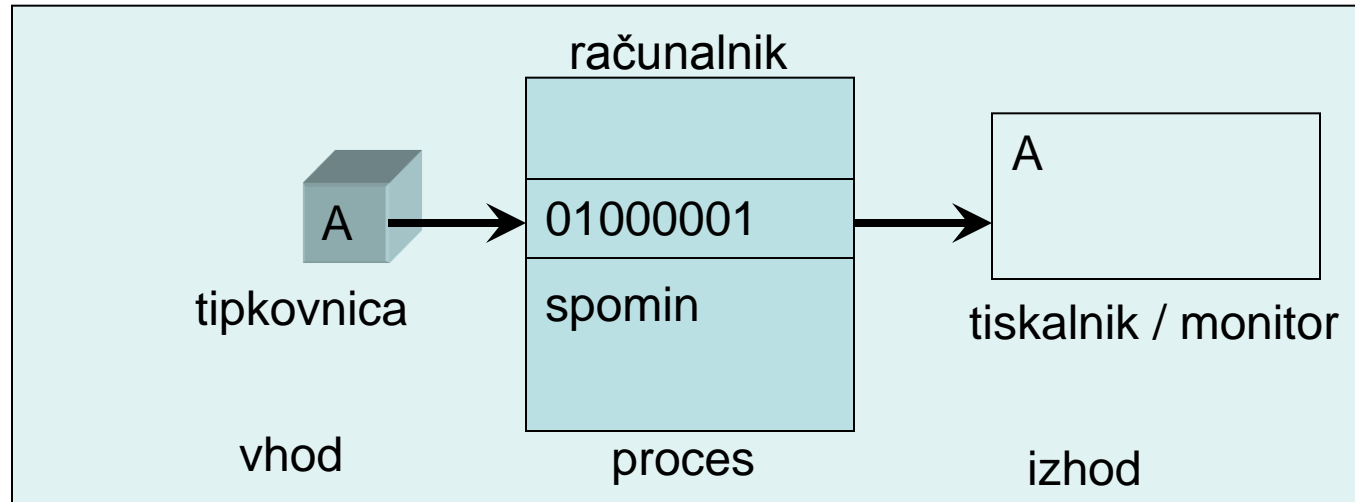
- programski jeziki
 - načrtovanje programske opreme
 - programiranje je proces ko dani problem formuliramo tako, da ga računalnik lahko izvede.
 - računalniški program je niz ukazov
-

Ali so vsi programi že napisani?

- Računalniki so prisotni 50 let. Pričakovali bi, da so vsi programi že napisani.
 - Na voljo je na stotine programov: programi za obdelavo besedila, računovodstvo, risanje, igre, načrtovanje, ...
 - Program:
 - kupim
 - kupim/dobim in nadgradim
 - napišem sam
-

Vhod / izhod

- Tipkovnica prinese revolucijo



- ? sodobni vmesniki

Vhodne enote

- tipkovnica
- miška
- zaslon na dotik
- pisalna plošča
- magnetno črnilo
- optični čitalci
- senzorji
- kamere
- Mikrofon
- ...

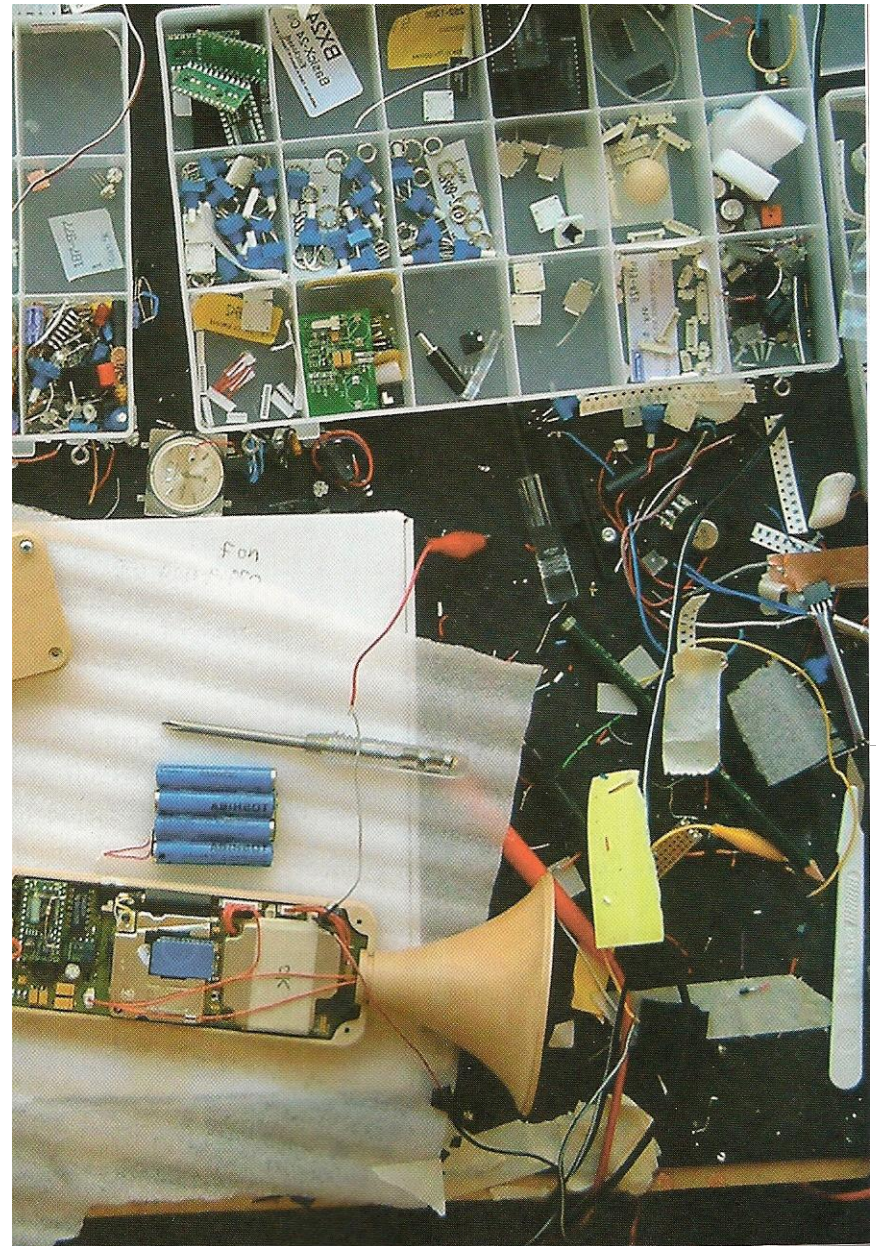
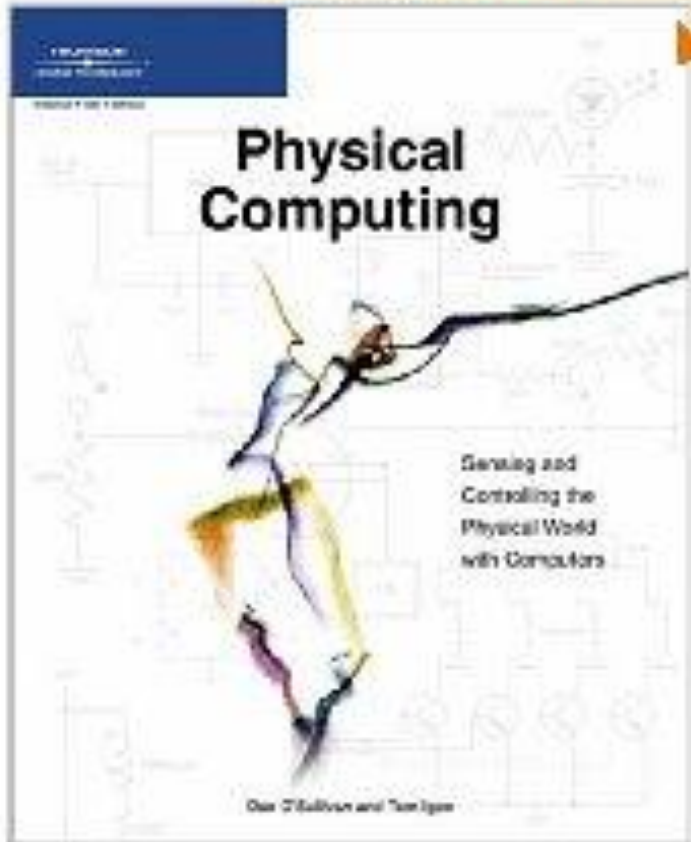


Izhodne enote

- Avdio/video prikazovalniki
- Tiskalniki, ploterji, mikrofilm, ...
- Disk, tračne enote, optične enote
- Brezžične in žične povezave



Click to **LOOK INSIDE!**





LED T-Shirt.COM


Processing.org

Processing


Cover \ Exhibition \ Reference \ Learning \ Download \ Shop \ About

» Feed » Forum » Wiki » Code


» Exhibition




[unnamed soundsculpture](#)
by Daniel Franke & Cedric Kiefer



[Soundmachines](#)
by The Product



[Composition No. 1](#)
by Visual Editions



[Max Planck Research Networks](#)
by Moritz Stefaner and Christopher Warnow

» Download Processing
» Play with Examples
» Browse Tutorials

Processing is an open source programming language and environment for people who want to create images, animations, and interactions. Initially developed to serve as a software sketchbook and to teach fundamentals of computer programming within a visual context, Processing also has evolved into a tool for generating finished professional work. Today, there are tens of thousands of students, artists, designers, researchers, and hobbyists who use Processing for learning, prototyping, and production.

- » Free to download and open source
- » Interactive programs using 2D, 3D or PDF output
- » OpenGL integration for accelerated 3D
- » For GNU/Linux, Mac OS X, and Windows
- » Projects run online or as double-clickable applications
- » Over 100 libraries extend the software into sound, video, computer vision, and more...
- » Well [documented](#), with many [books](#) available

To see more of what people are doing with Processing, check out these sites:

- » [Processing Wiki](#)
- » [Processing Discussion Forum](#)
- » [OpenProcessing](#)
- » [CreativeApplications.Net](#)
- » [O'Reilly Answers](#)
- » [Vimeo](#)
- » [del.icio.us](#)
- » [Flickr](#)

To contribute to the development please visit [Processing.org](#)

Arduino - HomePage

Main Site | Blog | Playground | Forum | Labs | Store

Help | Sign in or Register

ARDUINO

Buy | Download | Getting Started | Learning | Reference | Hardware | FAQ



Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the **Arduino programming language** (based on **Wiring**) and the Arduino development environment (based on **Processing**). Arduino projects can be stand-alone or they can communicate with software running on a computer (e.g. Flash, Processing, MaxMSP).

The boards can be **built by hand** or **purchased** preassembled; the software can be **downloaded** for free. The hardware reference designs (CAD files) are **available** under an open-source license, you are free to **adapt them to your needs**.

Arduino received an Honorary Mention in the Digital Communities section of the 2006 Ars Electronica

Photo by the Arduino Team

[Arduino on Twitter](#) ([more](#))
8 days ago
[@Shields_Arduino](#) The Arduino Due runs Arduino code

Pristopi k reševanju problema

1. Identifikacija problema
 2. Razumevanje problema
 3. Iskanje alternativnih poti za rešitev problema
 4. Izbor optimalne rešitve
 5. Nabor korakov za izvedbo
 6. Evalvacija rešitve
-

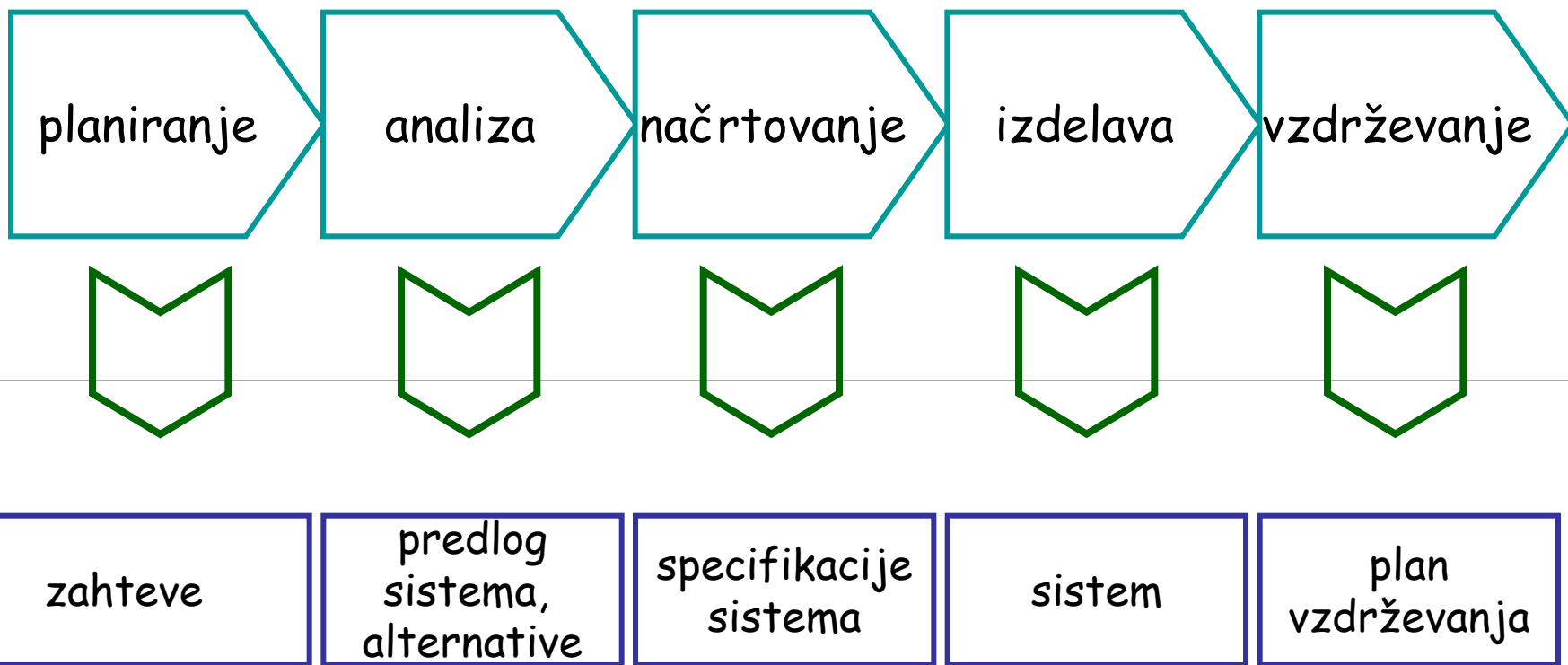
Pristopi k reševanju problemov

- top-down
 - analiza in dekompozicija problema
 - razgradnja sistema na podsisteme, modularnost
 - proceduralno programiranje: jezik C
 - bottom-up
 - sinteza
 - poudarek na kodiranju in testiranju
 - uporaba delov kode
 - objektno programiranje: jezika C++ in Java
 - običajno uporabimo kombinacijo obeh
 - UML (universal modelling language)
-

Razvoj programske opreme

1. Planiranje
 2. Analiza problema
 3. Načrtovanje rešitve
 4. Implementacija rešitve
 5. Vzdrževanje
-

SDLC (system development life cycle)



Primer LOC

Year	Operating System	SLOC (Million)
1993	Windows NT 3.1	4-5
1994	Windows NT 3.5	7-8
1996	Windows NT 4.0	11-12
2000	Windows 2000	more than 29
2001	Windows XP	40
2003	Windows Server 2003	50

LOC

Operating System	SLOC (Million)
Red Hat Linux 6.2	17
Red Hat Linux 7.1	30
Debian 2.2	55-59
Debian 3.0	104
Debian 3.1	215
Debian 4.0	283
OpenSolaris	9.7
FreeBSD	8.8
Mac OS X 10.4	86
Linux kernel 2.6.0	5.2

Planiranje rešitve

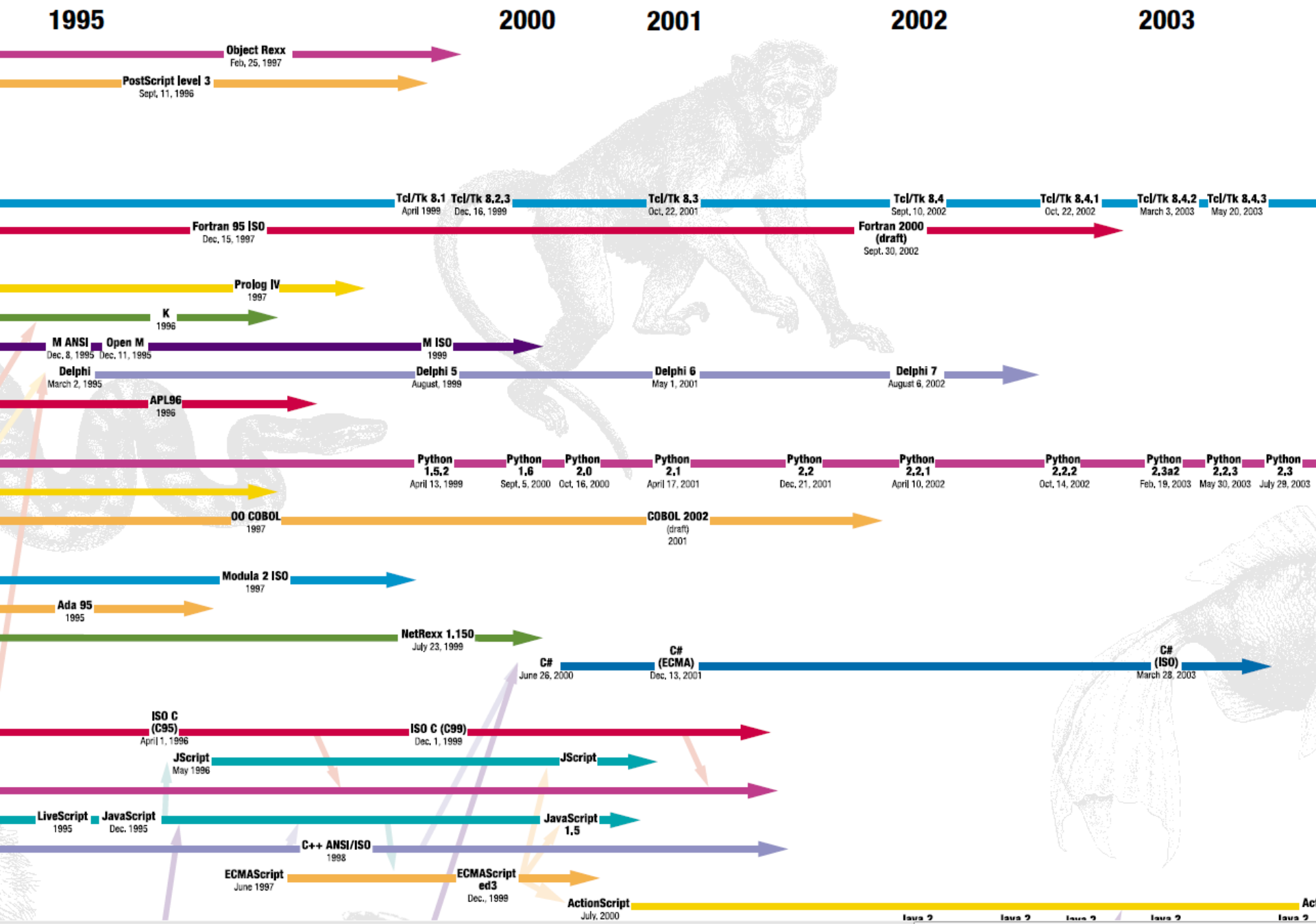
- Analiza problema
 - Vhodni podatki
 - Željeni rezultati
 - Zahtevana obdelava (procesiranje)
 - Nabor možnih rešitev

 - Komunikacija z računalnikom: Kodiranje

 - Plan testiranja
-

Principi načrtovanja

- analiza
 - načrtovanje, oblikovanje
 - testiranje, ocenjevanje
 - izboljšave
-
- programer $\leftarrow \rightarrow$ končni uporabnik
-



Programski jeziki

- programski jeziki:
 - proceduralni jeziki: C, Pascal, ...
 - objektno usmerjeni jeziki: C++, Java

 - skriptni jeziki: JavaScript
 - označevalni jeziki: HTML

 - ? izbor programskega jezika
 - Primeri uporabe programskih jezikov
-

Programski jeziki

- Proceduralno
 - Zgradba: glavni program (rutina), ki kliče funkcije
 - Reševanje sekvenčnih problemov
 - Neprimerno za interaktivne aplikacije

 - Objektno
 - Zgradba: objektni principi, kjer so objekti v interakciji med seboj – posnemanje realnega sveta
 - Rešuje probleme sekvenčnega programiranja
-

LOC (Lines of code)

C

```
#include <stdio.h>
int main(void) {
printf("Hello World");
return 0;
}
```

Lines of code: 5
(excluding whitespace)

COBOL

```
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. HELLOWORLD.
000300
000400*
000500 ENVIRONMENT DIVISION.
000600 CONFIGURATION SECTION.
000700 SOURCE-COMPUTER. RM-COBOL. 000800
OBJECT-COMPUTER. RM-COBOL. 000900
001000 DATA DIVISION.
001100 FILE SECTION.
001200
100000 PROCEDURE DIVISION.
100100
100200 MAIN-LOGIC SECTION.
100300 BEGIN.
100400 DISPLAY " " LINE 1 POSITION 1 ERASE EOS.
100500 DISPLAY "Hello world!" LINE 15 POSITION 10.
100600 STOP RUN.
100700 MAIN-LOGIC-EXIT.
100800 EXIT.
```

Lines of code: 17
(excluding whitespace)

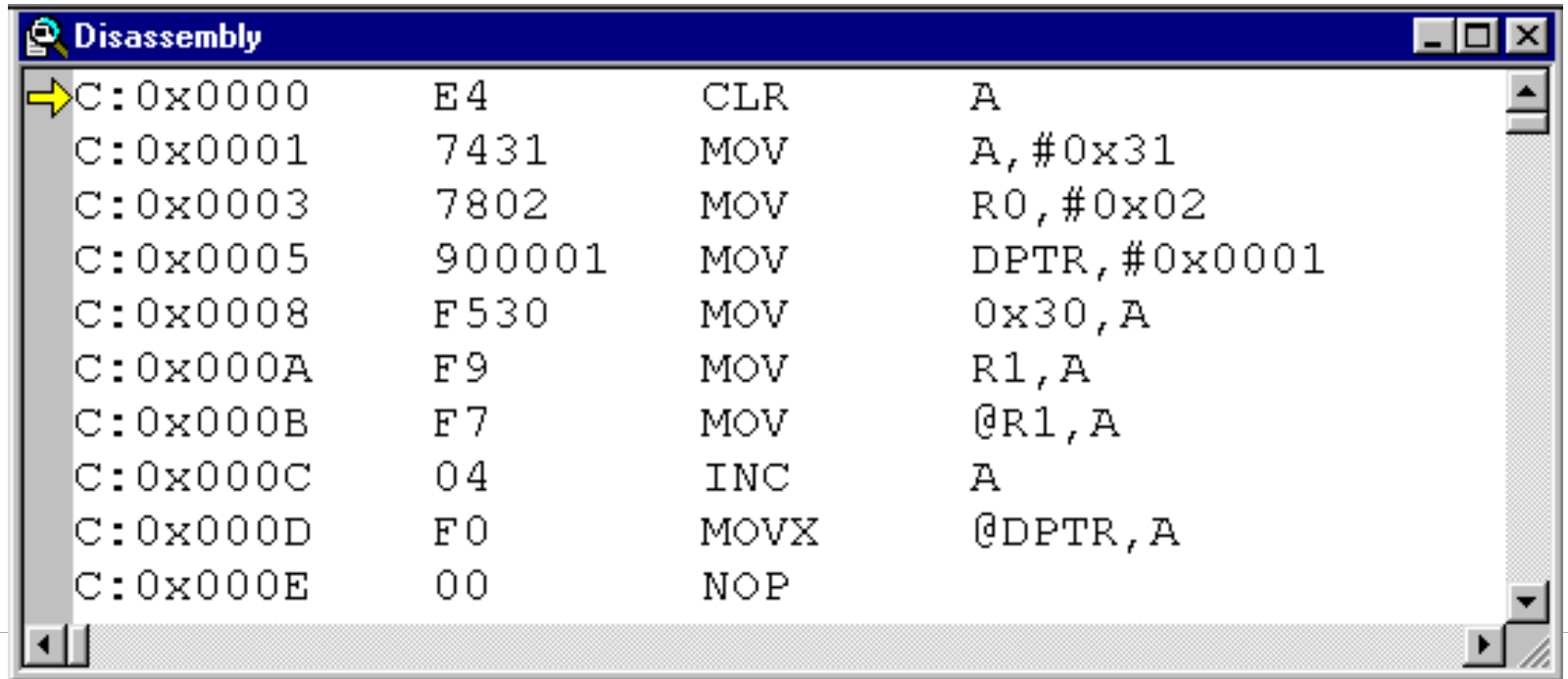
Programski jezik Java

- objektno orientiran jezik
- neodvisnost od platforme – operacijskega sistema
- izhodišče C++

```
public class Pozdrav {  
    public static void main(String[] args) {  
        System.out.println("Pozdravljeni!");  
    }  
}
```

Strojni jezik in zbirnik sta odvisna določena za dani CPE

- Sled preprostega programa



The screenshot shows a window titled "Disassembly" with a list of assembly instructions. The first instruction is highlighted with a yellow arrow. The instructions are as follows:

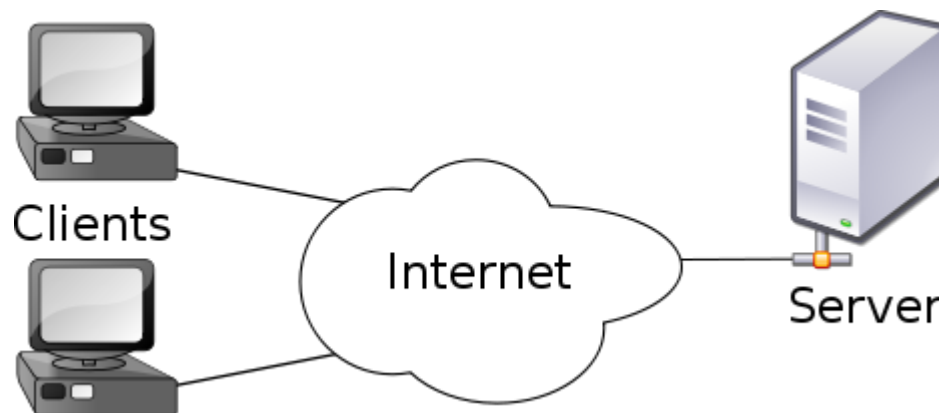
Address	Hex	Instruction	Comment
C:0x0000	E4	CLR	A
C:0x0001	7431	MOV	A, #0x31
C:0x0003	7802	MOV	R0, #0x02
C:0x0005	900001	MOV	DPTR, #0x0001
C:0x0008	F530	MOV	0x30, A
C:0x000A	F9	MOV	R1, A
C:0x000B	F7	MOV	@R1, A
C:0x000C	04	INC	A
C:0x000D	F0	MOVX	@DPTR, A
C:0x000E	00	NOP	

Programski jezik Java

- Principi objektnega programiranja
 - Definiranje razredov in objektov
 - Določitev interakcije med objekti
 - Načrtovanje programa in testiranje
 - Prevajalnik (compiler) prevede javansko kodo v strojno kodo
 - (JavaScript uporablja interpreter (intepreter)!)
-

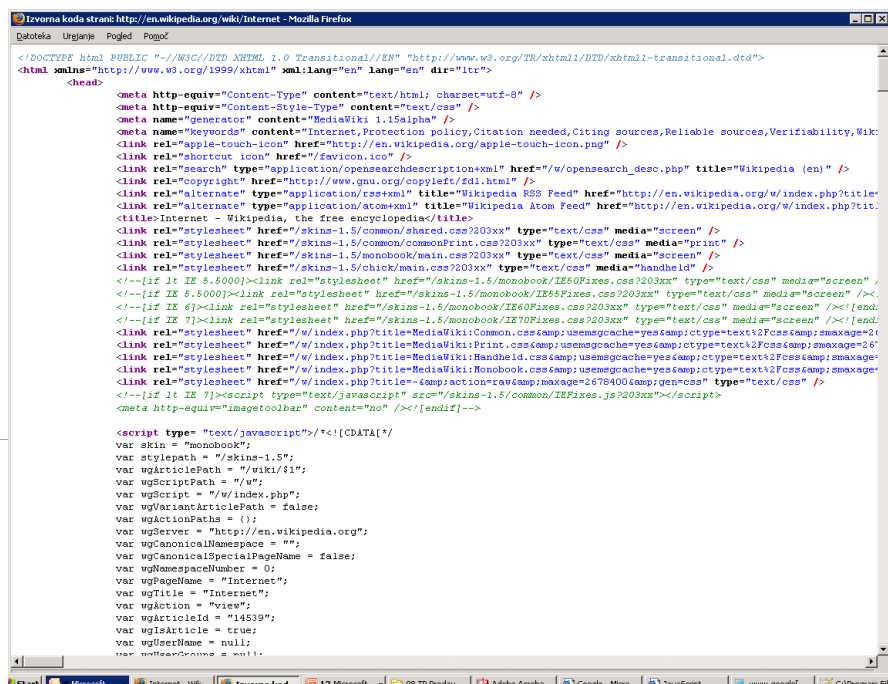
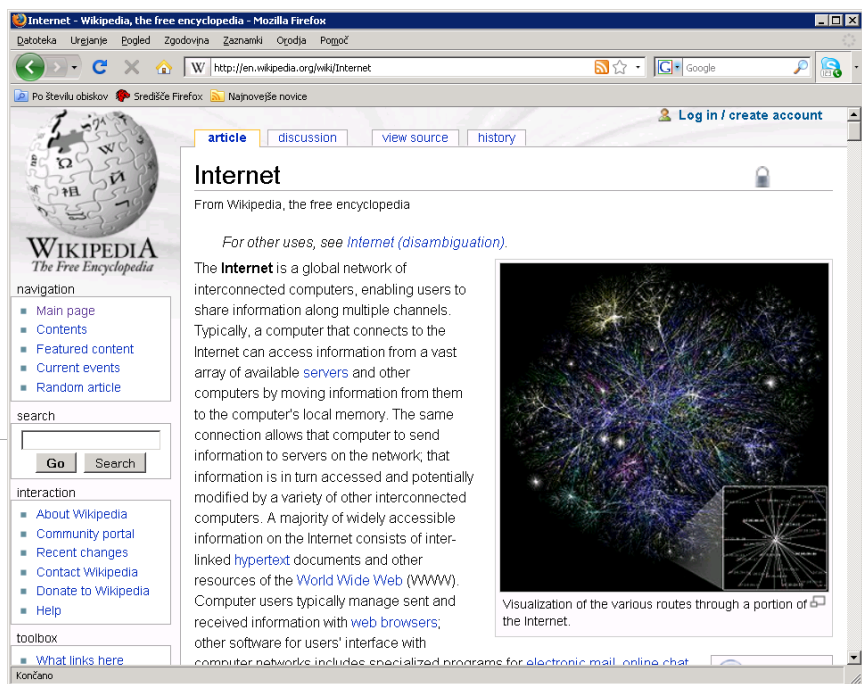
Odjemalec - Strežnik

- Programska koda na **odjemalcu (client)**:
 - Izvede se na odjemalcu, običajno del HTML datoteke
 - Primer: skripta napisana v jeziku JavaScript
 - Uporaba: dinamična vsebina spletne strani
- Programska koda na **strežniku (server)**:
 - Izvede se na strežniku pred pošiljanjem odgovora odjemalcu
 - Primer: Skripta napisana v jeziku PHP
 - Uporaba: generacija vsebine spletne strani s trenutnimi podatki v podatkovni bazi



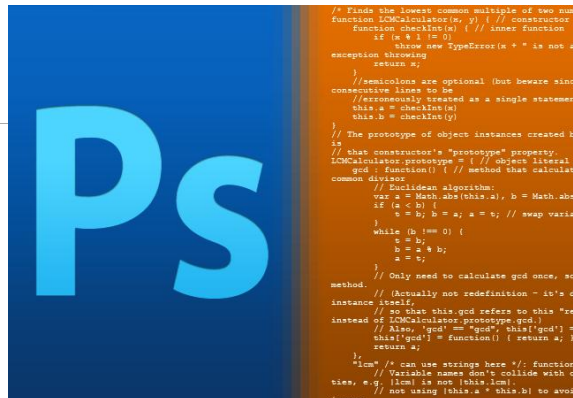
Spletne strani

- spletne strani sestavljajo: besedilo, slike, povezave, ...
- vsebino spletne strani lahko vidim, če v brskalniku pritisnem desno tipko in izberem View Source



JavaScript v Adobe Photoshop CS

- Uporaba skript za obdelavo gradiv v Adobe Photoshop okolju
 - Avtomatizacija obdelave velike količine slik
 - Prilagoditev in dodajanje ukazov
 - Dodajanje novih funkcionalnosti, prilagoditev uporabniškega vmesnika
- Možnost izvajanja odločitev in pogojev: pogojni in ponavljalni stavki



<http://www.howtogeek.com/howto/34334/how-to-use-javascript-to-save-time-by-automating-photoshop/>

<http://morris-photographics.com/photoshop/tutorials/scripting2.html>

<http://www.photoshopsupport.com/tutorials/jennifer/photoshop-scripts.html>

10 new HTML5 tags you need to know about

By [Justin James](#)

May 13, 2012, 8:51 AM PDT

Takeaway: HTML5 offers new tags and attributes that provide more power, efficiency, and flexibility for your Web development. Here are 10 tags you'll want to check out.

HTML5 brings a host of new elements and attributes to allow developers to make their documents more easily understood by other systems (especially search engines!), display data more uniquely, and take on some of the load that has required complex JavaScript or browser plug-ins like Flash and Silverlight to handle. Here are 10 new items in HTML5 that will make it easier for you to write your Web sites.

1: `<video>` and `<audio>`

One of the biggest uses for Flash, Silverlight, and similar technologies is to get a multimedia item to play. With HTML5 supporting the new video and audio controls, those technologies are now relegated to being used for fallback status. The browser can now natively display the controls, and the content can be manipulated through JavaScript. Don't let the codec confusion scare you away. You can specify multiple sources for content, so you can make sure that your multimedia will play regardless of what codecs the user's browser supports.

2: `<input>` type attributes

The venerable `<input>` element now has a number of new values for the type attribute, and browsers do some pretty slick things depending on its value. For example, set type to "datetime" and browsers can show calendar/clock controls to pick the right time, a trick that used to require JavaScript. There is a wide variety of type attributes, and learning them (and the additional attributes that go with some of them) will eliminate the need for a lot of JavaScript work.

3: `<canvas>`

<http://blog.jonasbandi.net/2012/02/future-smells-like-javascript.html>

THURSDAY, FEBRUARY 2, 2012

➔ The future smells like JavaScript

Of course I am only [repeating what others](#) are preaching about the recent [rise of JavaScript](#).

But I think the movement is significant and can't be overstated. And recent developments are really even making it more and more interesting.



Nobody can deny that [JavaScript](#) is the de facto programming language of Html5. Every other language trying to bolt itself onto Html5 looks like pure friction so far. And Html5 is looking upon a prospering future.

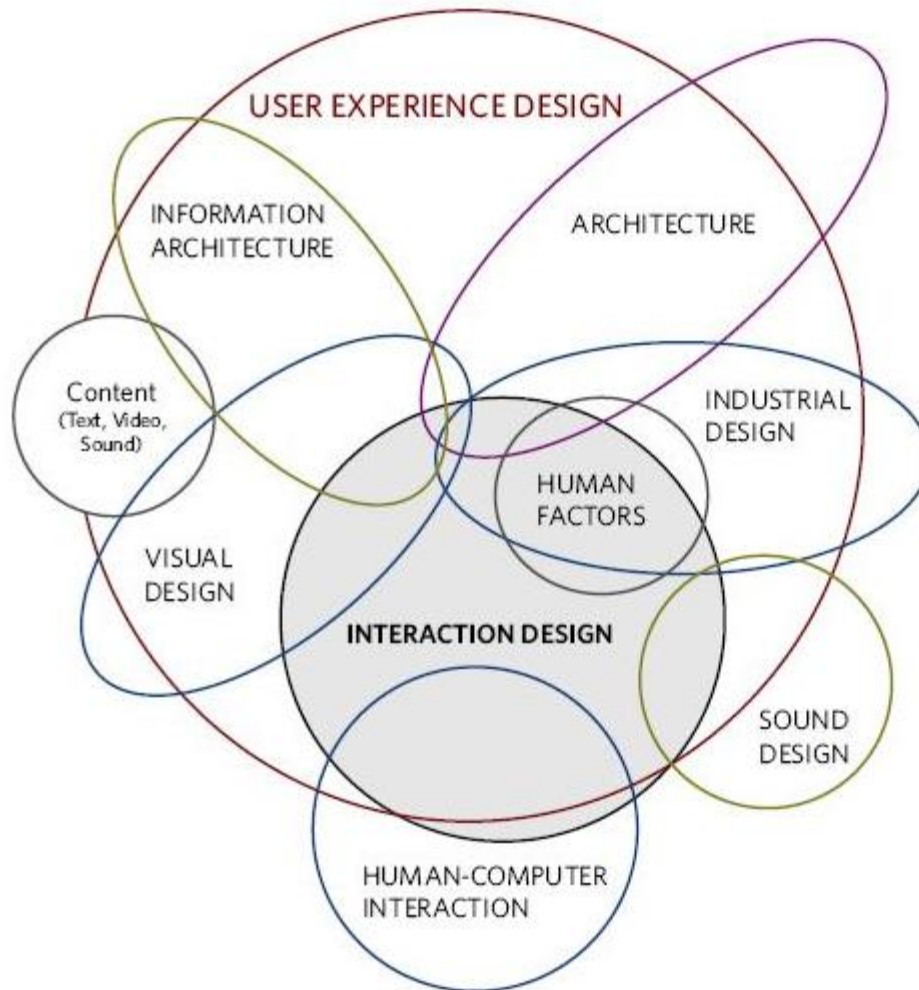


Figure 1.15

The disciplines surrounding interaction design.

